Information Technology Course
Module Software Engineering
by Damir Dobric / Andreas Pech

Approve Prediction of Multisequence Learning

Babar Ali (1439780)
ali.babar@stud.fra-uas.de

Muhammad Ahsan Tanveer (1449055)
muhmmad.tanveer@stud.fra-uas.de

Fazal Waheed (1430446)
fazal.waheed@stud.fra-uas.de

***Abstract*— This paper is about a technique called Multisequence Learning, which helps computer learn and predict sequence of things. We look at how this technique is currently being used and propose a new way to make it better. Our new method automates the process of reading sequences from a .csv file and testing predictions, which saves time and reduces mistakes compared to doing it manually. We show how this can be useful in different industries, like power consumption predictions or cancer related predictions. By testing our method, we found it accurately predicts sequences, making it useful for many applications. Overall, our paper explains Multisequence Learning and introduces a new method to make sequence prediction faster and more reliable.**

Keywords—Hierarchical Temporal Memory (HTM), Prediction code, Local Area Density, SDR, AI, HTM Prediction Engine).

INTRODUCTION

Multi-sequence learning with HTM means teaching the computer to spot and guess patterns in countless sets of information. We start by turning the data into special codes called Sparse Distributed Representations (SDRs) using something called a scaler encoder. Once the data is coded, another part called the spatial pooler makes simple version of the input sets, which are then sent to the temporal memory part to learn and guess patterns. Using HTM for multisequence learning is a strong way to find and guess patterns in many different sets of Information.

Hierarchical Temporal Memory (HTM) is a special kind of computer model made to imitate how our brain´s neocortex works. Scientists developed it to understand how our brain recognizes patterns. HTM has shown that its good at recognizing pattern in data, especially when it comes to understanding the order and movement of information over time.

In this project we have implemented new method along Multisequence Algorithm which can read dataset from .csv file for training the model. When the model is fully trained and learning is completed then it fetches the datasets for testing and predicting the next elements with Accuracy. In this project we have performed this learning approach with predicting the next elements on three scenarios the cancer prediction, power consumption and heart disease prediction with accuracy. [1]

LITERATURE SURVEY

A. Multisequence Learning

Multi-sequence learning is a machine learning paradigm that aims to learn from many data sequences at the same time. These sequences could be of many different forms, including time-series data, text documents, biological sequences (such as DNA or protein sequences), or any other sequential data. The basic purpose of multi-sequence learning is to model the dependencies and relationships inside and between multiple sequences to perform a task such as prediction, classification, generation, and clustering. This method is especially beneficial when there are interdependencies or correlations between multiple sequences that can offer context or information to the learning process.

Recurrent neural networks (RNNs), convolutional neural networks (CNNs) transformers, and various attention mechanisms are examples of common multi-sequence learning techniques. These models are capable of accurately expressing sequential patterns and interdependence over several sequences. C# multi-sequence learning can be implemented utilizing several tools and frameworks that allow sequence modeling and machine learning tasks.

B. Hierarchical Temporal Memory

HTM is a biologically inspired method that is based on the architecture of the neocortex and attempts to simulate how the human brain processes information about vision, hearing, behavior, and so on, ultimately leading to memory and prediction. An HTM system is trained rather than programmed; the basic learning techniques utilized are not limited to specific sensory domains and can be used to a wide range of challenges involving the modeling of complicated sensory input.HTM has come a long way from its humble beginnings to its current success. An HTM network is made up of hierarchically arranged regions with neurons organized in columns. The functional principle is captured in two algorithms, which are detailed in the original whitepaper [Hawkins et al., 2011].

In a few iterations, HTM analyses a distinct pattern and compares it to the preceding ones. To enable the training of diverse input pattern sequences that can be predicted, it is crucial that the input patterns are unique and not repetitive. [2]

Parameters	Default value
inputBits	100
numColumns	1024
CellsPerColumn	25
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	0.02 * numColumns
PotentialRadius	0.15 * inputBits
MaxBoost	10.0
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	0.02 * numColumns
Activation Threshold	15
Connected Permanence	0.5
Permanence Decrement	0.25
Permanence Increment	0.15
Predicted Segment Decrement	0.1

Table.1: HTM Config parameters

C. Sparse Distributed Representation

One of the most intriguing AI difficulties is the problem of knowledge representation. Traditional computer science methodologies make it challenging to represent everyday facts and relationships in a computer-readable format. The fundamental issue is that our understanding of the world is not organized into discrete facts with clear relationships. Traditional computer data formats cannot accurately represent the complex and ill-defined links between concepts. Brains do not have this issue. They represent datausing a technique known as Sparse Distributed Representations,or SDRs.

SDRs and their mathematical features are critical to biological intelligence. This book's principles are founded on SDRs, which underpin all brain functions. SDRs function as the brain's language. Human intelligence's adaptability and inventiveness are inextricably linked to this manner of representation. To achieve similar flexibility and creativity, intelligent robots should use the same representation mechanism, SDRs. An SDR is made up of hundreds of bits with just a small percentage being 1s and the remainder being 0. The bits in an SDR correspond to neurons in the brain, a 1 being a relatively active neuron and a 0 being a relatively inactive neuron. The most important property of SDRs is that each bit has meaning. [3]

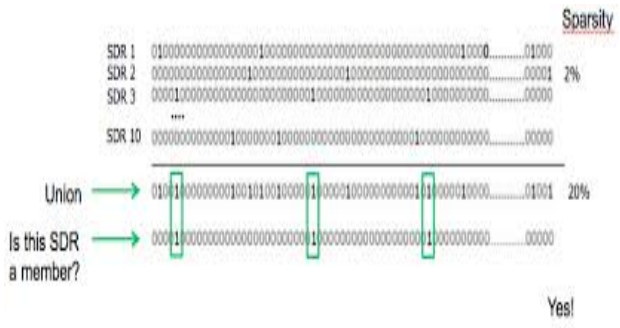

Fig.1: SDR Representation

D. Neocortex

The neocortex plays a crucial role in facilitating advanced cognitive processes, sensory interpretation, and intricate motor coordination within the brain. This part of the brain is primarily associated with higher-order functions, encompassing a wide range of mental activities such as thinking, perception, and complex movement. The cortex exhibits distinct structural features at the cellular level, attributed to the arrangement and makeup of different cell types and neural circuits, leading to region-specific differences. These variations in the cortex's cytoarchitecture play a crucial role in defining the functionality and specialization of different brain regions.

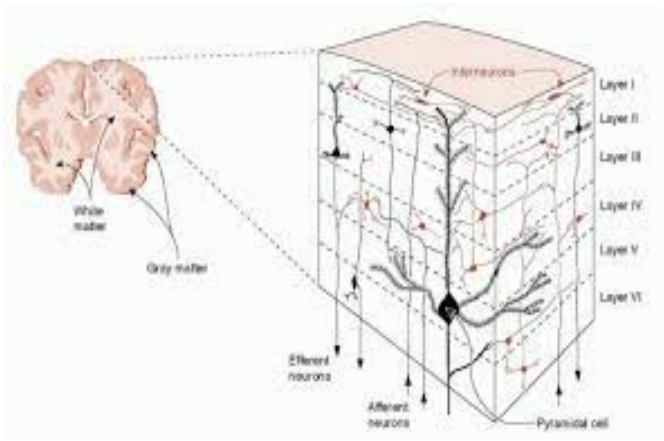

Fig.2: Neocortex Layers

In the realm of cognitive science, the Hierarchical Temporal Memory (HTM) stands as a conceptual framework designed to mirror various structural and algorithmic aspects observed in the neocortex, as elucidated by Hawkins et al. in 2011. HTM networks boast the capability to grasp time-ordered sequences through a continuous real-time learning process.HTM networks have displayed exceptional prowess in the domains of anomaly detection and sequence prediction when dealing with data streams, attaining cutting-edge performance levels. [4]

E. Memory

Memory is commonly perceived as a system responsible for processing information, encompassing both explicit and implicit functions. It comprises distinct components such as a sensory processor, short-term (or working) memory, and long-term memory. This concept can be linked to the functioning of neurons, indicating a complex interplay between memory processes and neurobiology.

F. Prediction

The neocortex functions as a virtual fortune teller, continually generating predictions based on what it knows and learns. It uses past experiences (code logic, data structures, etc.) and fresh inputs to predict what will happen next in a program. This predictive power is like having a crystal ball for code, allowing developers to anticipate problems, optimize performance, and design more resilient software. Just as we utilize prior experience to make educated

predictions about the future, the C# neocortex leverages its acquired information to foresee outcomes in programming tasks. Mastering how the neocortex anticipates in C# allows programmers to develop more efficient code and design programs that anticipate user wants. [5]

METHODOLOGY

The project multi-sequence learning, which uses Microsoft Visual Studio 2022 IDE (integrated development environment), is used as a reference model to describe the operation of multi-sequence learning, which uses the HTM prediction engine. This project aims to understand multi-sequence learning for the sequence of numbers and to develop a new approach that automates the process of reading learning sequences from a file and testing subsequence from another file to determine the percentage prediction accuracy based on the learning in HTM.

G. Datasets

A dataset is a systematically arranged assortment of information. It encompasses data gathered from observations, measurements, research, or analysis. Data comprises details like factual details, numerical data, statistics, names, and even rudimentary depictions of entities. This organized accumulation of data serves as a valuable resource for various purposes. By structuring information in a coherent manner, datasets facilitate efficient data management and analysis. Understanding the significance of data sets is crucial in fields such as research, analytics, and decision-making processes. This section describes the reference to the datasets which are used for Multi Sequence Learning.

Multiple sequence of numbers and alphabetic in .csv file for three different scenario. Figure 3 & 4 shows the testing and training data sets for power consumption prediction.

High	10	12	15	20	25	30	35	40	45	50
Low	60	66	68	70	73	76	79	82	84	86
Normal	88	89	90	92	94	95	96	97	98	48
Normal	10	15	18	25	30	35	40	45	50	63
Low	62	65	66	72	75	80	83	85	88	90
High	32	34	36	38	43	44	45	50	53	55

Fig:3 Training dataset for power consumption scenario.

10	12	15	20
92	94	95	96
65	66	72	75

Fig:4 Testing dataset for power consumption scenario.

H. Encoder

SDR inputs are the only ones that the HTM network accepts.It must have an encoder in order to be used for practical applications.The scalar encoder is the HTM system's initial and most important encoder
The principles of SDR encoding:

- SDRs with overlapping bits should be produced from semantically related material. The degree of resemblance increases with the overlap.
- It must be deterministic since the same input should always result in the same output.
- The output should have the same dimensions for all inputs.
- The output should have the same dimensions for all inputs.

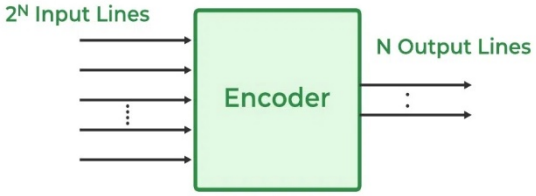

Fig.5: Digital Encoder [6]

I. Spatiol pooler

The initial layer of the HTM network is called the Spatial Pooler. It receives the encoder's SDR input and outputs a set of active columns. These columns compete with one another to be activated, signifying the input's recognition. The Spatial Pooler has two tasks: it must maintain the overlap qualities of the encoder's output and a fixed sparsity. These characteristics bear resemblance to normalization in other neural networks,which facilitates training by limiting neuronal activity.
Three versions of SP are implemented and considered:

- The original single-threaded Spatial Pooler version without any modifications to the method.
- Version of SP-MT that supports several cores on a single system in a multithreaded fashion.
- SP-Parallel, which facilitates multicore and multimode spatial pooler calculus.

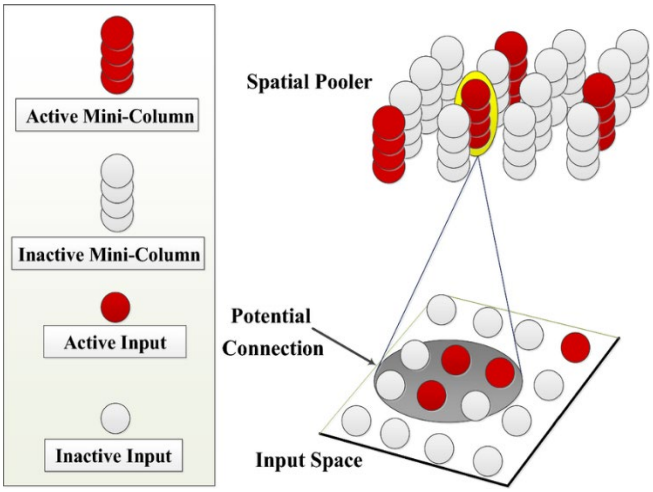

Fig.6: Spatial pooler-algorithm [6]

J. *Temporal Memory*

Temporal memory in HTM is in charge of memorizing sequences of patterns throughout time. It accomplishes this by creating hierarchical connections between neurons, which encode both spatial and temporal patterns. Temporal memory is a notion in artificial intelligence and neuroscience that is most commonly connected with Numenta's Hierarchical Temporal Memory (HTM) model. It's a process that mimics the neocortex's temporal memory capacities, which are responsible for higher cognitive activities. In the HTM model, temporal memory allows the system to learn sequences and generate predictions based on sequences of patterns it encounters. Temporal memory in HTM models is critical for tasks that require processing time-varying data, such as natural language comprehension, anomaly detection, and sensor data analysis. By capturing the temporal connections inherent in sequential data, temporal memory allows HTM systems to make accurate predictions and comprehend complex patterns across time. [6]

NEW METHOD CREATED IN HELPER.CS

Line no: 16 – 68
```
public static List<Dictionary<string, string>>
ReadSequencesDataFromCSV(string dataFilePath)
```

Line no: 120 – 161
```
public static EncoderBase FetchEncoder(int predictionScenario)
```

Line no: 206 – 229
```
Public static List<string> ReadTestingSequencesDataFromCSV(string dataFilePath)
```

Line no: 237 - 244
```
Public static int[] EncodeTestingElement(string testingElement, int predictionScenario)
```

Line no: 251 - 300
```
public static List<Dictionary<string, string>>ReadSequencesDataFromCSVForPC(
```

CLASS CREATED

Figure 7 shows the flow chart how this complete experiment and accuracy is calculated.

MainProgram.cs
ProjectStarter.cs

NEW METHOD CREATED IN
CancerPredictionTrainingAndTesting.cs
HeartDiseasePredictionTrainingAndTesting.cs
PowerConsumptionPredictionTrainingAndTesting.cs

Line no: 14 - 46
```
public void RunMultiSequenceLearningExperiment(int predictionScenario, string trainingDataFilePath, string testingDataFilePath)
```

Line no: 55 - 158
```
private static double PredictElementAccuracy(Predictor predictor, string sequenceData, int predictionScenario)
```

RESULTS

In this project we have covered three cases for multi-sequence learning. We concluded the experiment as many times as possible using various datasets. To minimize execution time, we kept the dataset size and the number of sequences small.

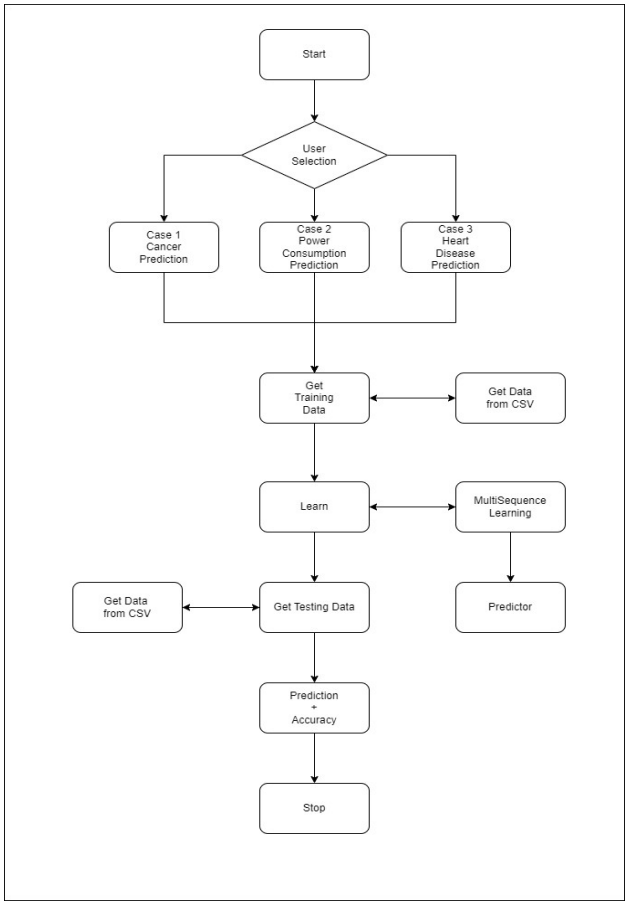

Fig.7: Flow chart for Multi Sequence Learning Experiment

Result for Cancer Prediction

Figure 8 shows the Prediction of next elements with accuracy for Cancer Prediction scenario. For example, sequence UJBUUBUJ is active for cancer. The model predicted next elements with 100% accuracy.

```
Ready to Predict.....
------------------------------
Sequence UJBUUBUJ
------------------------------
Element U
Predicted next element: J
Element J
Predicted next element: B
Element B
Predicted next element: U
Element U
Predicted next element: U
Element U
Predicted next element: B
Element B
Predicted next element: U
Element U
Predicted next element: J
------------------------------
Sequence UJBUUBUJ
Predicted Mode : mod. active
Accuracy 100%
------------------------------
Sequence GGLGLGFF
------------------------------
Element G
Predicted next element: G
Element G
Predicted next element: L
Element L
Predicted next element: G
Element G
Predicted next element: L
Element L
Predicted next element: G
Element G
Predicted next element: F
Element F
Predicted next element: F
------------------------------
Sequence GGLGLGFF
Predicted Mode : very active
Accuracy 100%
------------------------------
Sequence AKWAKAAK
------------------------------
Element A
Predicted next element: K
Element K
Predicted next element: W
Element W
Predicted next element: A
Element A
Predicted next element: K
Element K
Predicted next element: A
Element A
Predicted next element: A
Element A
Predicted next element: K
------------------------------
Sequence AKWAKAAK
Predicted Mode : inactive - virtual
Accuracy 100%
```

Fig.8: Prediction of next elements with accuracy for CancerPrediction scenario

Result for Power Consumption Prediction

Figure 9 shows the *Prediction of next elements with accuracy for* Power Consumption *scenario.*

```
------------------TRAINING END----
Ready to Predict.....
Sequence 10,12,15,20
------------------------------
Element 10
Predicted next element: 12
Element 12
Predicted next element: 15
Element 15
Predicted next element: 20
------------------------------
Power Consumption : High
Accuracy 100%
Sequence 92,94,95,96
------------------------------
Element 92
Predicted next element: 94
Element 94
Predicted next element: 95
Element 95
Predicted next element: 96
------------------------------
Power Consumption : Normal
Accuracy 100%
Sequence 65,66,72,75
------------------------------
Element 65
Predicted next element: 66
Element 66
Predicted next element: 72
Element 72
Predicted next element: 75
------------------------------
Power Consumption : Low
Accuracy 100%
```

Fig.9: Prediction of next elements with accuracy for power consumption scenario

Result for Heart Disease Prediction

Figure 10 shows the *Prediction of next elements with accuracy for* Heart Disease Prediction *scenario.*

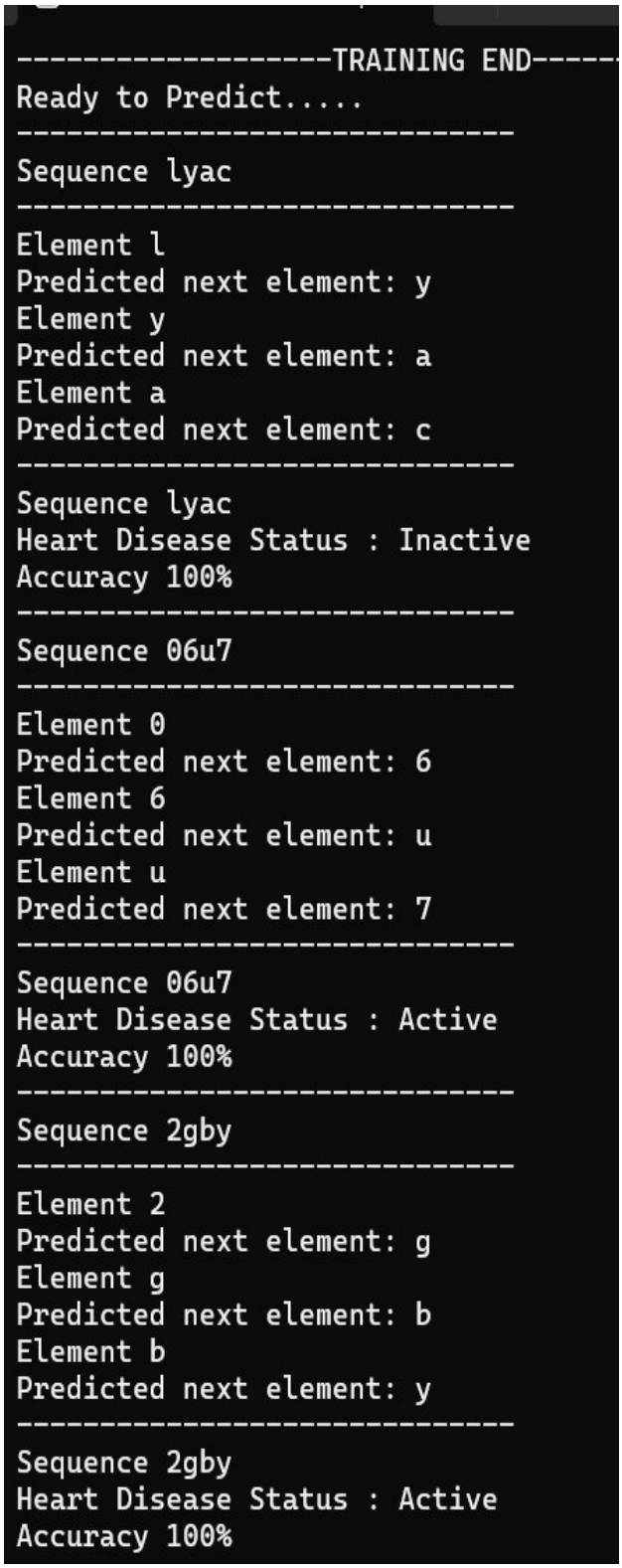

Fig.10: Prediction of next elements with accuracy for heart disease scenario

REFERENCES

[1] H. J. George D, “Towards a Mathematical Theory of Cortical Micro-circuits,” Numenta Inc, CA, 2009.

[2] T. L. A. ,. M. Regina Sousa, "Hierarchical Temporal Memory Theory Approach to Stock Market Time Series Forecasting," *MDPI,* vol. 10, no. 14, 08 JULY 2021.

[3] P. Kanerva, Sparse Distrubuted Memory, Cambridge, Mass. : MIT Press, 1998, p. 155.

[4] B. G. t. Csongor Pilinszki-Nagy1, "Performance Analysis of Sparse Matrix Represtation in Hierarchical Temporal Memory for Sequence Modeling," *INFOCOMMUNICATIONS JOURNAL,* vol. 12, p. 50, 2020.

[5] P. A. Simona Lodato, "Generating Neuronal Diversity in the Mammalian Cerebral Cortex," *HHS Author Manuscripts,* 2015.

[6] m. R. a. H. shiva Sanati, "Performence Camparison of Different HTM-Spatial Pooler Algorithms Based on Information-Theoratic Measures," Feb 2024.

[7] D. Dobric, "Neocortex," [Online]. Available: https://github.com/ddobric/neocortexapi.